(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2004-245988
(P2004-245988A)
(43)公開日　平成16年9月2日(2004.9.2)

(51)Int.Cl.[7]	F I		テーマコード（参考）
G09C 1/00	G09C	1/00 610A	5J104

審査請求 未請求　請求項の数 12　OL　（全 12 頁）

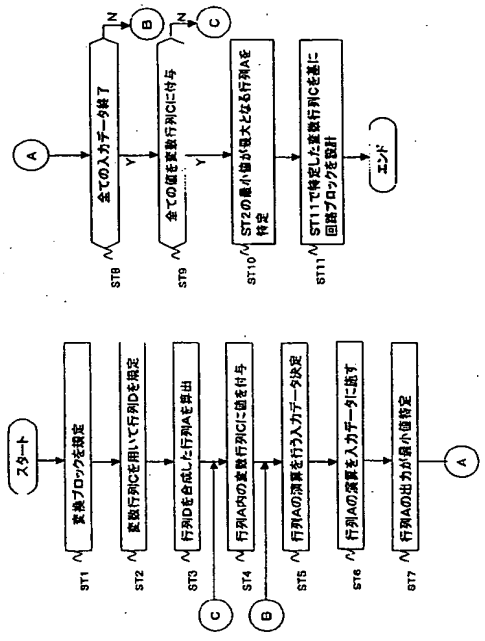
(21)出願番号　特願2003-34591 (P2003-34591)
(22)出願日　平成15年2月13日 (2003.2.13)

(71)出願人　000002185
ソニー株式会社
東京都品川区北品川６丁目７番３５号
(74)代理人　100094053
弁理士　佐藤　隆久
(72)発明者　金丸　昌司
東京都品川区北品川６丁目７番３５号　ソニー株式会社内

Ｆターム(参考)　5J104 JA05

(54)【発明の名称】データ処理装置、その方法およびそのプログラムと線形変換回路および暗号化回路

(57)【要約】
【課題】アクティブＳ－ｂｏｘ数を最大にする線形変換を特定するデータ処理方法を提供する。
【解決手段】複数の線形変換候補のうち、当該線形変換候補を実現する回路上の制約を満たす複数の前記線形変換候補を特定し（ＳＴ１，ＳＴ２，ＳＴ３）、当該特定した線形変換候補のそれぞれについて複数の入力データを基に線形変換処理を行い、それらの処理結果内に生じる零の数の最小値（いわゆるアクティブＳ－ｂｏｘ）を求め（ＳＴ４～ＳＴ９）、その最小値を最大とする線形変換候補を特定する（ＳＴ１０）。そして、当該特定した線形変換候補を基に、線形変換部を構成する（ＳＴ１１）。
【選択図】　図４



【特許請求の範囲】
【請求項１】
線形変換を実現する回路上の制約をそれぞれ満たす複数の線形変換候補を特定する第１の工程と、
前記第１の工程で特定した前記複数の線形変換候補のそれぞれについて、複数の入力データに対して当該線形変換候補が規定する線形変換を行ったそれぞれの結果に零が生じる個数の最小値を特定する第２の工程と、
前記第１の工程で特定した前記複数の線形変換候補のうち、前記第２の工程で特定した前記最小値が最大となる前記線形変換候補を特定する第３の工程と
を有するデータ処理方法。
【請求項２】
前記第１の工程は、前記複数の線形変換候補として、置換行列内の２つの零領域のうち一方を変数行列で置き換えた複数の単位線形変換の合成である線形変換を特定し、
前記第２の工程は、前記複数の単位線形変換の前記変数行列として異なる複数の行列を付与して得られる前記線形変換候補の各々について、前記最小値を特定する
請求項１に記載のデータ処理方法。
【請求項３】
前記複数の単位線形変換の数がＭであり、
前記単位線形変換は、Ｍ行Ｍ列の行列演算で実現する
請求項２に記載のデータ処理方法。
【請求項４】
前記第１の工程は、前記線型変換候補を、変数行列C_1，C_2，C_3，C_4を用いて下記式（１）で規定する
請求項３に記載のデータ処理方法。
【数１】

$$\begin{pmatrix} I + C_4C_3 + C_2C_1 + C_4C_3C_2C_1 + C_4C_1 & C_2 + C_4C_3C_2 + C_4 \\ C_3 + C_3C_2C_1 + C_1 & I + C_3C_2 \end{pmatrix} \quad \cdots (1)$$

【請求項５】
前記線形変換が、共通鍵ブロック暗号のラウンド関数処理内で規定された線形変換である場合に、
前記第２の工程は、平文データを非線形拡散処理して得られた前記入力データに対して前記線形変換を行う
請求項１に記載のデータ処理方法。
【請求項６】
前記第３の工程で特定した前記線形変換候補に対応する前記単位線形変換を実現する回路ブロックを有する線形変換回路を構成する第４の工程
をさらに有する請求項１に記載のデータ処理方法。
【請求項７】
コンピュータによって実行されるプログラムであって、
線形変換を実現する回路上の制約をそれぞれ満たす複数の線形変換候補を特定する第１の手順と、
前記第１の手順で特定した前記複数の線形変換候補のそれぞれについて、複数の入力データに対して当該線形変換候補が規定する線形変換を行ったそれぞれの結果に零が生じる個数の最小値を特定する第２の手順と、
前記第１の手順で特定した前記複数の線形変換候補のうち、前記第２の手順で特定した前記最小値が最大となる前記線形変換候補を特定する第３の手順と

を記述したプログラム。
【請求項８】
前記第１の手順は、前記複数の線形変換候補として、置換行列内の２つの零領域のうち一方を変数行列で置き換えた複数の単位線形変換の合成である線形変換を特定し、
前記第２の手順は、前記複数の単位線形変換の前記変数行列として異なる複数の行列を付与して得られる前記線形変換候補の各々について、前記最小値を特定する
請求項７に記載のプログラム。
【請求項９】
線形変換を実現する回路上の制約をそれぞれ満たす複数の線形変換候補を特定する第１の手段と、
前記第１の手段で特定した前記複数の線形変換候補のそれぞれについて、複数の入力データに対して当該線形変換候補が規定する線形変換を行ったそれぞれの結果に零が生じる個数の最小値を特定する第２の手段と、
前記第１の手段で特定した前記複数の線形変換候補のうち、前記第２の手段で特定した前記最小値が最大となる前記線形変換候補を特定する第３の手段と
を有するデータ処理装置。
【請求項１０】
共通鍵ブロック暗号のラウンド関数処理内で規定された線形変換を行う線形変換回路であって、
複数のデータのそれぞれに対応した複数のデータ線と、
前記複数のデータ線を介して入力された前記複数のデータに線形変換を順に施す複数の回路ブロックと
を有し、
前記回路ブロックの各々は、前記複数のデータ線のうち一部の複数の前記データ線上に設けられた演算回路を有し、演算回路が設けられていない各々の前記データ線から最大１個の前記演算回路に対してデータが供給されるように構成されている
線形変換回路。
【請求項１１】
前記回路ブロックの各々は、前記複数のデータ線を同数の第１のデータ線群と第２のデータ線群とに分けた場合に、前記第１のデータ線群内の前記データ線にのみ前記演算回路を有し、前記第２のデータ線群内の前記データ線から、前記第１のデータ線群内の前記演算回路にデータを出力するように構成されている
請求項１０に記載の線形変換回路。
【請求項１２】
ラウンド関数処理を行って共通鍵ブロック暗号化を行う暗号化装置であって、
前記ラウンド関数処理内で規定された非線形変換を行う非線形変換回路と、
前記非線形変換回路により前記非線形変換が施された入力データに対して線形変換を行う線形変換回路と
を有し、
前記線形変換回路は、
前記入力データを構成する複数のデータのそれぞれに対応した複数のデータ線と、
前記データ線を介して入力された前記複数のデータに線形変換を順に施す複数の回路ブロックと
を有し、
前記回路ブロックの各々は、前記複数のデータ線のうち一部の複数の前記データ線上に設けられた演算回路を有し、演算回路が設けられていない各々の前記データ線から最大１個の前記演算回路に対してデータが供給されるように構成されている
暗号化装置。
【発明の詳細な説明】
【０００１】

【発明の属する技術分野】
本発明は、暗号化処理などで規定された線形変換を行う線形変換回路の設計に用いられるデータ処理方法、装置、そのプログラムと、線形変換回路および暗号化装置に関する。
【０００２】
【従来の技術】
情報セキュリティを達成するために種々の暗号化技術開発されている。
このような暗号化技術の一種である共通鍵ブロック暗号は、例えば、非線形処理と線形処理（拡散処理）とからなるラウンド関数を規定している。
上記ラウンド関数の非線形処理は、Ｓ－ｂｏｘと呼ばれるユニットで構成され、入出力間の非線形性を実現している。
また、上記ラウンド関数の線形処理は、多ビットからなる入力データの影響を複数ビットに拡散させる線形変換を行う。
このような線形変換を用いる方法として、ＡＥＳ（Ａｄｖａｎｃｅｄ Ｅｎｃｒｙｐｔｉｏｎ Ｓｔａｎｄａｒｄ）等で用いられるＭＤＳ（Ｍａｘｉｍａｌ Ｄｉｓｔａｎｃｅ Ｓｅｐａｒａｂｌｅ）を利用したものがある。
ＭＤＳは、ＧＦ（2^8）等の拡大体上の変換を用いることによって、効率よくビット拡散を行う手法である。
しかしながら、ＭＤＳは、実装時に回路構成が複雑になるという欠点がある。
このような決定を解消するものとして、ＣａｍｅｌｌｉａおよびＥ２などの暗号手法がある。この暗号手法では、高速かつ小規模な構成の回路を構成するために、ＧＦ（２）上の変換が用いられている。
【０００３】
【特許文献１】
特開２００２－９１２９５号公報
【０００４】
【発明が解決しようとする課題】
しかしながら、ＧＦ（２）上の変換による高い拡散効率を得るために、いわゆるアクティブ（Ａｃｔｉｖｅ）Ｓ－ｂｏｘ数を最大にする回路構成を、線形変換を実現する回路上の制約とは無関係に、全ての線形変換候補について演算を行って決定しており、膨大な計算量が必要になるという問題がある。
ここで、アクティブＳ－ｂｏｘ数は、複数の入力データに対して上記ラウンド関数の線形処理を行い、それらの処理結果内に生じる零の数の最小値である。
【０００５】
本発明はかかる事情に鑑みてなされたものであり、その目的は、複数の線形変換候補のなかから、複数の入力データに線形変換を行った結果に零が生じる個数の最小値が最大となる線形変換候補を従来に比べて少ない演算量で特定できるデータ処理方法、その装置および、そのプログラムを提供することを目的とする。
また、本発明は、上述したデータ処理方法、その装置およびそのプログラムによって設計される線形変換回路および暗号化装置を提供することを目的とする。
【０００６】
【課題を解決するための手段】
上述した目的を達成するために、第１の発明のデータ処理方法は、線形変換を実現する回路上の制約をそれぞれ満たす複数の線形変換候補を特定する第１の工程と、前記第１の工程で特定した前記複数の線形変換候補のそれぞれについて、複数の入力データに対して当該線形変換候補が規定する線形変換を行ったそれぞれの結果に零が生じる個数の最小値を特定する第２の工程と、前記第１の工程で特定した前記複数の線形変換候補のうち、前記第２の工程で特定した前記最小値が最大となる前記線形変換候補を特定する第３の工程とを有する。
【０００７】
第１の発明のデータ処理方法の作用は以下のようになる。

先ず、第１の工程において、線形変換を実現する回路上の制約をそれぞれ満たす複数の線形変換候補を特定する。
次に、第２の工程において、前記第１の工程で特定した前記複数の線形変換候補のそれぞれについて、複数の入力データに対して当該線形変換候補が規定する線形変換を行ったそれぞれの結果に零が生じる個数の最小値を特定する。
次に、第３の工程において、前記第１の工程で特定した前記複数の線形変換候補のうち、前記第２の工程で特定した前記最小値が最大となる前記線形変換候補を特定する。

【０００８】

第１の発明のデータ処理方法は、好ましくは、前記第１の工程は、前記複数の線形変換候補として、置換行列内の２つの零領域のうち一方を変数行列で置き換えた複数の単位線形変換の合成である線形変換を特定し、前記第２の工程は、前記複数の単位線形変換の前記変数行列として異なる複数の行列を付与して得られる前記線形変換候補の各々について、前記最小値を特定する。

【０００９】

第２の発明のプログラムは、コンピュータによって実行されるプログラムであって、線形変換を実現する回路上の制約をそれぞれ満たす複数の線形変換候補を特定する第１の手順と、前記第１の手順で特定した前記複数の線形変換候補のそれぞれについて、複数の入力データに対して当該線形変換候補が規定する線形変換を行ったそれぞれの結果に零が生じる個数の最小値を特定する第２の手順と、前記第１の手順で特定した前記複数の線形変換候補のうち、前記第２の手順で特定した前記最小値が最大となる前記線形変換候補を特定する第３の手順とを記述している。

【００１０】

第２の発明のプログラムは、コンピュータによって実行され、前述した第１の発明の各工程を実現する。

【００１１】

第３の発明のデータ処理装置は、線形変換を実現する回路上の制約をそれぞれ満たす複数の線形変換候補を特定する第１の手段と、前記第１の手段で特定した前記複数の線形変換候補のそれぞれについて、複数の入力データに対して当該線形変換候補が規定する線形変換を行ったそれぞれの結果に零が生じる個数の最小値を特定する第２の手段と、前記第１の手段で特定した前記複数の線形変換候補のうち、前記第２の手段で特定した前記最小値が最大となる前記線形変換候補を特定する第３の手段とを有する。

【００１２】

第３の発明のデータ処理装置の作用は以下のようになる。
先ず、第１の手段において、線形変換を実現する回路上の制約をそれぞれ満たす複数の線形変換候補を特定する。
次に、第２の手段において、前記第１の手段で特定した前記複数の線形変換候補のそれぞれについて、複数の入力データに対して当該線形変換候補が規定する線形変換を行ったそれぞれの結果に零が生じる個数の最小値を特定する。
次に、第３の手段において、前記第１の手段で特定した前記複数の線形変換候補のうち、前記第２の手段で特定した前記最小値が最大となる前記線形変換候補を特定する。

【００１３】

第４の発明の線形変換回路は、共通鍵ブロック暗号のラウンド関数処理内で規定された線形変換を行う線形変換回路であって、複数のデータのそれぞれに対応した複数のデータ線と、前記複数のデータ線を介して入力された前記複数のデータに線形変換を順に施す複数の回路ブロックとを有し、前記回路ブロックの各々は、前記複数のデータ線のうち一部の複数の前記データ線上に設けられた演算回路を有し、演算回路が設けられていない各々の前記データ線から最大１個の前記演算回路に対してデータが供給されるように構成されている。

【００１４】

第５の発明の暗号化装置は、ラウンド関数処理を行って共通鍵ブロック暗号化を行う暗号

化装置であって、前記ラウンド関数処理内で規定された非線形変換を行う非線形変換回路と、前記非線形変換回路により前記非線形変換が施された入力データに対して線形変換を行う線形変換回路とを有し、前記線形変換回路は、前記入力データを構成する複数のデータのそれぞれに対応した複数のデータ線と、前記データ線を介して入力された前記複数のデータに線形変換を順に施す複数の回路ブロックとを有し、前記回路ブロックの各々は、前記複数のデータ線のうち一部の複数の前記データ線上に設けられた演算回路を有し、演算回路が設けられていない各々の前記データ線から最大１個の前記演算回路に対してデータが供給されるように構成されている。

【００１５】

【発明の実施の形態】

以下、本発明の実施形態に係わる回路設計方法および暗号化装置について説明する。

先ず、本実施形態の回路設計方法によって設計される線形変換回路を組み込んだ暗号化装置について説明する。

図１は、本実施形態の暗号化装置１の構成図である。

暗号化装置１は第５の発明の暗号化装置に対応している。

図１に示すように、暗号化装置１は、例えば、鍵生成回路２、初期処理回路３、Ｎ個のＦｅｉｓｔｅｌ構造モジュール４＿１～４＿Ｎおよび後処理回路５を有する。ここで、Ｎは２以上の整数である。

暗号化装置１は、共通鍵ブロック暗号を行う。

【００１６】

鍵生成回路２は、鍵データＫ１，Ｋ２＿１～Ｋ２＿Ｎ，Ｋ３を生成し、鍵データＫ１を初期処理回路３に出力し、鍵データＫ２＿１～Ｋ２＿ＮをそれぞれＦｅｉｓｔｅｌ構造モジュール４＿１～４＿Ｎに出力し、鍵データＫ３を後処理回路５に出力する。

【００１７】

初期処理回路３は、入力した平文データＰＴに初期変換を施してデータＳ３を生成する。

データＳ３は、例えば、１２８ビットのデータである。

初期処理回路３は、データＳ３の下位６４ビットのデータＳ３ａをＦｅｉｓｔｅｌ構造モジュール４＿１のＦ関数回路１１に出力し、下位６４ビットのデータＳ３ｂをＦｅｉｓｔｅｌ構造モジュール４＿１のＸＯＲ（ｅＸｃｌｕｓｉｖｅ　ＯＲ：排他的論理和）回路１２に出力する。

【００１８】

Ｆｅｉｓｔｅｌ構造モジュール４＿１～４＿Ｎの各々は、Ｆ関数回路１１およびＸＯＲ回路１２を有する。

Ｆｅｉｓｔｅｌ構造モジュール４＿１～４＿Ｎは、直列に接続され、同じ構成を有している。

以下、Ｆｅｉｓｔｅｌ構造モジュール４＿１について説明する。

Ｆ関数回路１１は、初期処理回路３からのデータＳ３ａに対して、非線形処理および線形処理（拡散処理）を施して、データＳ１１を生成し、これをＸＯＲ回路１２に出力する。

Ｆ関数回路１１は、本発明のラウンド関数処理を行う。

Ｆ関数回路１１の構成については、後に詳細に説明する。

【００１９】

ＸＯＲ回路１２は、初期処理回路３からのデータＳ３ｂとＦ関数回路１１からのデータＳ１１との排他的論理和を演算し、その結果であるデータＳ１２を後段のＦｅｉｓｔｅｌ構造モジュール４＿２のＦ関数回路１１に出力する。

また、Ｆ関数回路１１は、初期処理回路３からのデータＳ３ａを後段のＦｅｉｓｔｅｌ構造モジュール４＿２のＸＯＲ回路１２に出力するように構成されている。

その前段のＦｅｉｓｔｅｌ構造モジュール４＿Ｎ－１からのデータＳ３ａを下位６４ビットとし、Ｆｅｉｓｔｅｌ構造モジュール４＿ＮのＸＯＲ回路１２からのデータＳ１２を上位６４ビットとした１２８ビットのデータが、最終段のＦｅｉｓｔｅｌ構造モジュール４＿Ｎから後処理回路５に出力される。

【００２０】
後処理回路５は、Ｆｅｉｓｔｅｌ構造モジュール４＿Ｎからの１２８ビットのデータに対して、鍵生成回路２からの鍵データＫ３を用いて後処理を行い、その結果である暗号化データＣＴを出力する。
【００２１】
以下、図１に示すＦ関数回路１１の構成について説明する。
図２は、図１に示すＦ関数回路１１の構成図である。
図２に示すように、Ｆ関数回路１１は、例えば、ＸＯＲ部２１、非線形変換部２２、線形変換部２３、ＸＯＲ部２４および非線形変換部２５を有する。
ここで、非線形変換部２２が第５の発明の非線形変換回路に対応している。
また、線形変換部２３が第１の発明～第３の発明を用いた設計対象となり、第４および第５の発明の線形変換回路に対応している。
【００２２】
Ｆ関数回路１１では、入力された６４ビットのデータＳ３ａが、各々８ビットの８個のデータモジュールに分割されて処理される。
ＸＯＲ部２１は、データＳ３ａを分割して得られた８個のデータモジュールの各々に対して、鍵生成回路２から入力した鍵データＫ１との排他的論理和演算を施し、その結果をそれぞれ非線形変換部２２に出力する。
【００２３】
非線形変換部２２は、上記８個のデータモジュールに対応してそれぞれ設けられた非線形変換回路３１を有し、非線形変換回路３１において、入力したデータモジュールに対して非線形変換処理を施し、その結果である８個のデータモジュールを線形変換部２３に出力する。
非線形変換回路３１は、例えば、Ｓ－ｂｏｘと呼ばれる。
【００２４】
線形変換部２３は、非線形変換部２２からのデータをバイト単位でＧＦ（２）上の演算である拡散処理を行う。
線形変換部２３は、非線形変換部２２から入力した８個のデータモジュールをそれぞれ伝送する８個のデータ線２６＿１～２６＿８（第４および第５の発明のデータ線）を有する。
線形変換部２３は、図２に示すように、直列に接続された４個の回路ブロック４１＿１～４１＿４（第４および第５の発明の回路ブロック）を有する。
回路ブロック４１＿１は、データ線２６＿５～２６＿８上の各々にＸＯＲ回路（第４および第５の発明の演算回路）を配設している。
また、データ線２６＿１，２６＿２，２６＿３，２６＿４上のデータモジュールが、各々データ線２６＿５，２６＿６，２６＿７，２６＿８上のＸＯＲ回路に入力されるように配線されている。
回路ブロック４１＿２は、データ線２６＿１～２６＿４上の各々にＸＯＲ回路を配設している。
【００２５】
また、データ線２６＿５，２６＿６，２６＿７，２６＿８上のデータモジュールが、各々データ線２６＿３，２６＿４，２６＿１，２６＿２上のＸＯＲ回路に入力されるように配線されている。
また、データ線２６＿１，２６＿２，２６＿３，２６＿４上のデータモジュールが、各々データ線２６＿６，２６＿７，２６＿８，２６＿５上のＸＯＲ回路に入力されるように配線されている。
また、データ線２６＿５，２６＿６，２６＿７，２６＿８上のデータモジュールが、各々データ線２６＿１，２６＿２，２６＿３，２６＿４上のＸＯＲ回路に入力されるように配線されている。
【００２６】

ＸＯＲ部２４は、線形変換部２３から入力した８個のデータモジュールの各々に対して、鍵生成回路２から入力した鍵データＫ３との排他的論理和演算を施し、その結果をそれぞれ非線形変換部２５に出力する。

【００２７】

非線形変換部２５は、ＸＯＲ部２４からの上記８個のデータモジュールに対応してそれぞれ設けられた非線形変換回路３２を有し、非線形変換回路３２において、入力したデータモジュールに対して非線形変換処理を施し、その結果である８個のデータモジュールを出力する。

非線形変換回路３２は、例えば、Ｓ－ｂｏｘと呼ばれる。

非線形変換部２５から出力された８個のデータモジュールは、図２に示すように、組み合わされて図１に示すデータＳ１１としてＸＯＲ回路１２に出力される。

【００２８】

以下、図２に示すＦ関数回路１１の線形変換部２３の設計方法について説明する。

【００２９】

図３は、図２に示すＦ関数回路１１の線形変換部２３の設計に用いられるコンピュータ３９を説明するための図である。

図３に示すように、コンピュータ３９は、例えば、メモリ５１、操作部５２、ディスプレイ５３およびＣＰＵ５４を有し、これらがバス５０を介して接続されている。

ここで、コンピュータ３９が第３の発明のデータ処理装置に対応している。

メモリ５１は、コンピュータ３９が実行するプログラム４８（第２の発明のプログラム）、並びにコンピュータ３９によるプログラム４８の実行に用いられる種々のデータを記憶する。

操作部５２は、キーボードやマウス等であり、ユーザによる操作に応じた操作信号をＣＰＵ５４に出力する。

ディスプレイ５３は、コンピュータ３９の処理結果を表示する。

ＣＰＵ５４は、メモリ５１から読み出したプログラム５８を実行し、図２に示すＦ関数回路１１の設計処理を行う。

ＣＰＵ５４は、複数の線形変換候補のうち、当該線形変換候補を実現する回路上の制約を満たす複数の前記線形変換候補を特定し、当該特定した線形変換候補のそれぞれについて複数の入力データを基に線形変換処理を行い、それらの処理結果内に生じる零の数の最小値（いわゆるアクティブＳ－ｂｏｘ）を求め、その最小値を最大とする線形変換候補を特定する。そして、ＣＰＵ５４は、当該特定した線形変換候補を基に、図２に示す線形変換部２３を構成する。

【００３０】

以下、ＣＰＵ５４の設計処理手順（本実施形態の設計方法）を説明する。

図４は、ＣＰＵ５４の設計処理手順を説明するためのフローチャートである。

以下の手順の一部は、例えば、ＣＰＵ５４がディスプレイ５３に表示した操作画面を基にユーザが行った操作部５２の操作に応じて、ユーザとＣＰＵ５４との間で対話形式で行われる。

なお、図４に示すステップＳＴ１～ＳＴ３が第１の発明の第１の工程、第２の発明の第１の手順、並びに第３の発明の第１の手段に対応している。

ステップＳＴ４～ＳＴ９が第１の発明の第２の工程、第２の発明の第２の手順、並びに第３の発明の第２の手段に対応している。

ステップＳＴ１０が第１の発明の第３の工程、第２の発明の第３の手順、並びに第３の発明の第３の手段に対応している。

【００３１】

ステップＳＴ１：

ユーザは、設計対象の図２に示す線形変換部２３の線形変換を例えば、４つの線形変換ブロックに分割して規定し、その情報を操作部５２を介してＣＰＵ５４に与える。

【００３２】

ステップＳＴ２：
ＣＰＵ５４は、ステップＳＴ１で受けた情報を基に、下記式（２），（３），（４），（５）に示すように、各々４×４の行列である変数行列C_1，C_2，C_3，C_4を用いて、行列D_1，D_2，D_3，D_4を規定する。
このように、各々８×８の行列である行列D_1，D_2，D_3，D_4を規定することで、行列D_1，D_2，D_3，D_4を実現した図２に示す回路ブロック４１＿１～４１＿４を、ＸＯＲ回路が設けられていないデータ線から、最大１個のＸＯＲ回路にデータモジュールを出力するように構成できる。これにより、回路ブロック４１＿１～４１＿４に規定された回路上の制約が満たされる。
具体的には、図２に示すように、上位あるいは下位の４本のデータ線上にＸＯＲ回路を設け、当該ＸＯＲ回路が設けられていない４本のデータ線から、それぞれ異なる上記ＸＯＲ回路にデータモジュールを出力するように回路ブロック４１＿１～４１＿４を構成できる。
すなわち、本実施形態では、上述したように行列D_1～D_4を規定することで、回路上の制約を満たさないものについては、行列D_1～D_4の候補から予め除外することができる。

【００３３】
【数２】

$$D_1 = \begin{pmatrix} I & O \\ C_1 & I \end{pmatrix} \quad \cdots(2)$$

$$D_2 = \begin{pmatrix} I & C_2 \\ O & I \end{pmatrix} \quad \cdots(3)$$

$$D_3 = \begin{pmatrix} I & O \\ C_3 & I \end{pmatrix} \quad \cdots(4)$$

$$D_4 = \begin{pmatrix} I & C_4 \\ O & I \end{pmatrix} \quad \cdots(5)$$

【００３４】
ステップＳＴ３：
ＣＰＵ５４は、下記式（６），（７）に示すように、ステップＳＴ２で規定した行列D_1，D_2，D_3，D_4を合成した行列Ａ（本発明の線形変換候補）を算出する。
【００３５】
【数３】

$$A = D_1 D_2 D_3 D_4 \quad \cdots(6)$$

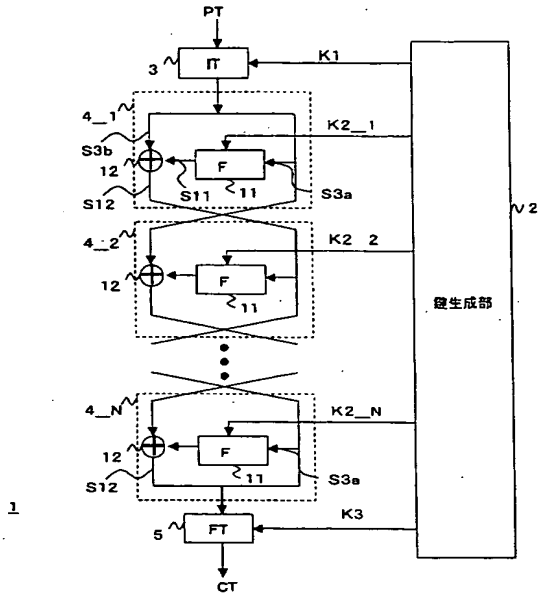
【００３６】
【数４】

$$A = \begin{pmatrix} I & C_4 \\ O & I \end{pmatrix}\begin{pmatrix} I & O \\ C_3 & I \end{pmatrix}\begin{pmatrix} I & C_2 \\ O & I \end{pmatrix}\begin{pmatrix} I & O \\ C_1 & I \end{pmatrix}$$
$$= \begin{pmatrix} I + C_4C_3 + C_2C_1 + C_4C_3C_2C_1 + C_4C_1 & C_2 + C_4C_3C_2 + C_4 \\ C_3 + C_3C_2C_1 + C_1 & I + C_3C_2 \end{pmatrix}$$
・・・(7)

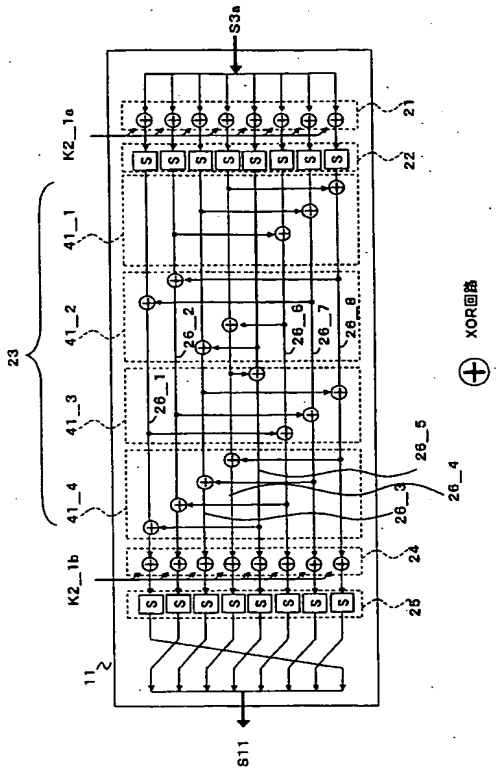
【００３７】
ステップＳＴ４：
ＣＰＵ５４は、ステップＳＴ３で生成した行列Ａ内の変数行列C_1，C_2，C_3，C_4の各要素に所定の値を与える。
これにより、本発明にいう、「複数の線形変換候補の特定」が行われる。
なお、ＣＰＵ５４は、ステップＳＴ９からのループバックにより当該ステップＳＴ４の処理を複数回行い、その度に、異なる行列Ａを規定する。
【００３８】
ステップＳＴ５：
ＣＰＵ５４は、予め決められた複数の入力データ（本発明の入力データ）のうち、次に行列Ａに入力する、すなわち行列Ａによる演算対象とする入力データを決定する。
なお、ＣＰＵ５４は、ステップＳＴ８からのループバックにより当該ステップＳＴ５の処理を複数回行い、その度に、異なる入力データを決定する。
【００３９】
ステップＳＴ６：
ＣＰＵ５４は、ステップＳＴ５で決定した入力データに対してステップＳＴ４で決定した行列Ａの演算を行う。
ステップＳＴ７：
ＣＰＵ５４は、ステップＳＴ６の演算結果（６４ビットデータ）内に含まれる零（０）の数を計数し、その計数値が、それまで計数した最小値より小さい場合に最小値を更新する。ＣＰＵ５４は、全ての行列Ａの各々について当該最小値（本発明の最小値）を求める。
【００４０】
ステップＳＴ８；
ＣＰＵ５４は、上記予め決められた全ての入力データについて、ステップＳＴ６の処理を行ったか否かを判断し、行ったと判断した場合にステップＳＴ９の処理に進み、そうでない場合にステップＳＴ５の処理に戻る。
ステップＳＴ９；
ＣＰＵ５４は、変数行列C_1，C_2，C_3，C_4を用いて規定可能な全ての行列Ａについて、ステップＳＴ６の処理を行ったか否かを判断し、行ったと判断した場合にステップＳＴ１０の処理に進み、そうでない場合にステップＳＴ４の処理に戻る。
【００４１】
ステップＳＴ１０：
ＣＰＵ５４は、ステップＳＴ２で最終的に得られた全ての行列Ａの最小値のうち、最大の最小値を出した行列Ａを特定する。
ステップＳＴ１１：
ＣＰＵ５４は、ステップＳＴ１０で特定した行列Ａで用いられた変数行列C_1，C_2，C_3，C_4を用いて図２示す線形変換部２３の回路ブロック４１＿１～４１＿４を構成（設計）する。
【００４２】
以上説明したように、本実施形態の設計方法によれば、複数の線形変換候補のうち、当該線形変換候補を実現する回路上の制約を満たす複数の前記線形変換候補を特定し、当該特定した線形変換候補のなかから、線形変換を行った結果に零が生じる個数の最小値を最大

にする線形演算候補を探索（特定）するため、全ての線形変換候補を対象としてを探索を行う場合に比べて、演算量を大幅に削減できる。
具体的には、本実施形態の設計方法によれば、変数行列C_1～C_4の各々は4×4の行列であるため、4！（4の階乗）通りの候補がある。従って、（C_1，C_2，C_3，C_4）の組み合わせは、$(4!)^4$通り、すなわち線形変換候補は、約2^{18}通りになる。
これに対して、従来では、GF（2）上の8×8行列の全てを線形変換候補するため、線形変換候補は2^{64}通りになる。
従って、本実施形態の設計方法によれば、設計に伴う演算量を従来に比べて大幅に削減できる。これにより、本実施形態によれば、線形変換部23の設計を実用的な時間で行うことが可能になる。
【0043】
本発明は上述した実施形態には限定されない。
上述した実施形態では、GF（2）上の変換をバイト単位で行い、32ビット演算を高速に行う場合を例示したが、本発明は、例えば、16ビットワードをGF（2）上の変換の単位として64ビット演算を行ったり、バイトをGF（2）上の変換の単位として64ビット演算を行うことで、上記線形変換を行うように設計を行ってもよい。
【0044】
【発明の効果】
以上説明したように、本発明によれば、複数の線形変換候補のなかから、複数の入力データに線形変換を行った結果に零が生じる個数の最小値が最大となる線形変換候補を従来に比べて少ない演算量で特定できるデータ処理方法、その装置および、そのプログラムを提供できる。
また、本発明によれば、上述した本発明のデータ処理方法、その装置およびそのプログラムによって設計される線形変換回路および暗号化装置を提供できる。
【図面の簡単な説明】
【図1】図1は、本発明の実施形態に係わる暗号化装置の構成図である。
【図2】図2は、図1に示すF関数回路の構成図である。
【図3】図3は、図2に示すF関数回路の線形変換部の設計に用いられるコンピュータを説明するための図である。
【図4】図4は、図3に示すCPUの設計処理手順を説明するためのフローチャートである。
【符号の説明】
1…暗号化装置、2…鍵生成回路、3…初期処理回路、4_1～4_N…Feistel構造モジュール、5…後処理回路、11…F関数回路、12…XOR回路、21…XOR部、22…非線形変換部、23…線形変換部、24…XOR部、25…非線形変換部、41_1～41_4…回路ブロック、39…コンピュータ、51…メモリ、52…操作部、53…ディスプレイ、54…CPU、58…プログラム

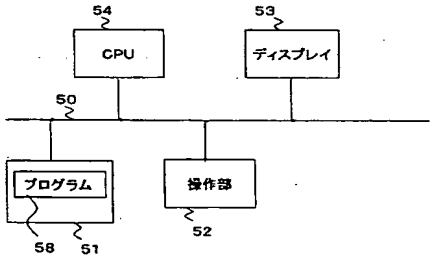
【図１】


PT
IT
K1
3
4_1
K2_1
S3b
12
F
S11
11
S3a
S12
4_2
K2_2
4_N
K2_N
鍵生成部
V2
FT
K3
5
CT
1

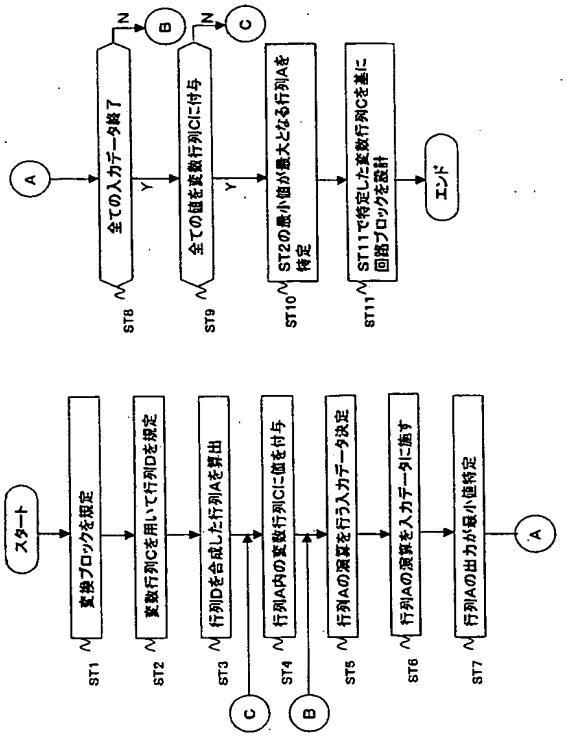
【図２】


S3a
K2_1a
K2_1b
41_1
41_2
41_3
41_4
23
26_1
26_2
26_3
26_4
26_5
26_6
26_7
26_8
21
22
24
25
11
S11
XOR回路

【図３】


54
CPU
53
ディスプレイ
50
プログラム
操作部
58
51
52
39

【図４】


スタート
変換ブロックを規定
ST1
変数行列Cを用いて行列Dを規定
ST2
行列Dを合成した行列Aを算出
ST3
行列A内の変数行列Cに値を付与
ST4
行列Aの演算を行う入力データ決定
ST5
行列Aの演算を入力データに当てる
ST6
行列Aの出力が最小値特定
ST7
A
B
C
全ての入力データ終了
ST8
全ての値を変数行列Cに付与
ST9
ST2の最小値が最大となる行列Aを特定
ST10
ST11で特定した変数行列Cを基に回路ブロックを設計
ST11
エンド
Y
N